

AMENDMENTS TO THE CLAIMS

Please amend Claims 1, 2, 13, 14, 25, and 26 as follows:

1 1. (currently amended) A method of securely invoking an access control function,
2 the method comprising the steps of:
3 receiving a digital signature for the access control function;
4 generating a mapping of the access control function to the digital signature;
5 determining that the digital signature is mapped to the access control function
6 based on the mapping when execution of the access control function is
7 requested;
8 determining whether an executable element matches the access control function
9 based on the digital signature; and
10 executing the executable element only when the executable element matches the
11 access control function[[]] ; and
12 wherein a particular class defines an implementation of the access control
13 function.

1 2. (currently amended) The method of Claim 1,
2 ~~wherein a particular class defines an implementation for the access control~~
3 ~~function;~~
4 wherein the step of receiving a digital signature includes the step of receiving a
5 digital signature for the particular class; and
6 wherein the step of generating a mapping includes generating a mapping between
7 the particular class and the digital signature.

1 3. (original) The method of Claim 1,

2 wherein the method further includes the step of detecting that an access control
3 event has occurred; and
4 wherein the step of retrieving the executable element is performed in response to
5 detecting that the event has occurred.

1 4. (original) The method of Claim 3,
2 wherein the method further includes the steps of:
3 generating a mapping between the access control event and the access
4 control function;
5 determining that the access control event is mapped to the access control
6 function; and
7 wherein the step of retrieving the executable element is performed in response to
8 determining that the access control event is mapped to the access control
9 function.

1 5. (original) The method of Claim 4, further including the step of the executable
2 element returning name-value pairs.

1 6. (original) The method of Claim 5, wherein the step of the executable element
2 returning name-value pairs includes the executable element returning a hash table
3 that contains the name-value pairs.

1 7. (original) The method of Claim 1, wherein the method further includes the
2 steps of:
3 generating a mapping of a plurality of access control functions to digital
4 signatures, wherein the plurality of access control functions include the
5 access control function, wherein one or more classes define an
6 implementation for each of the plurality of access control functions; and

7 wherein each of the one or more classes belong to a superclass.

1 8. (original) The method of Claim 7, further including the step of invoking a
2 routine defined by a superclass that collects data to return to a caller of the
3 particular class.

1 9. (original) The method of Claim 8, wherein the step of executing the
2 executable element includes invoking a routine defined for the superclass.

1 10. (original) The method of Claim 1, wherein the step of retrieving an
2 executable element includes retrieving byte code.

1 11. (original) The method of Claim 10, wherein the step of retrieving byte code
2 includes retrieving Java byte code.

1 12. (original) The method of Claim 1, wherein the step of retrieving an
2 executable element includes a first computer system retrieving byte code
3 transmitted via a local area network from a second computer system.

1 13. (currently amended) A computer-readable medium carrying one or more
2 sequences of one or more instructions for securely invoking an access control
3 function, the one or more sequences of one or more instructions including
4 instructions which, when executed by one or more processors, cause the one or
5 more processors to perform the steps of:

6 receiving a digital signature for the access control function;

7 generating a mapping of the access control function to the digital signature;

8 determining that the digital signature is mapped to the access control function

9 based on the mapping when execution of the access control function is

10 requested;

11 determining whether an executable element matches the access control function
12 based on the digital signature; ~~and~~
13 executing the executable element only when the executable element matches the
14 access control function[[]] ; and
15 wherein a particular class defines an implementation of the access control
16 function.

14. (currently amended) The computer-readable medium of Claim 13,
~~wherein a particular class defines an implementation for the access control~~
~~function;~~
wherein the step of receiving a digital signature includes the step of receiving a
digital signature for the particular class; and
wherein the step of generating a mapping includes generating a mapping between
the particular class and the digital signature.

15. (original) The computer-readable medium of Claim 13,
wherein the computer-readable medium further includes sequences of instructions
for performing the step of detecting that an access control event has
occurred; and
wherein the step of retrieving the executable element is performed in response to
detecting that the event has occurred.

16. (original) The computer-readable medium of Claim 15,
wherein the computer-readable medium further includes sequences of instructions
for performing the steps of:
generating a mapping between the access control event and the access
control function;

6 determining that the access control event is mapped to the access control
7 function; and
8 wherein the step of retrieving the executable element is performed in response to
9 determining that the access control event is mapped to the access control
10 function.

1 17. (original) The computer-readable medium of Claim 16, further including
2 sequences of instructions for performing the step of the executable element
3 returning name-value pairs.

1 18. (original) The computer-readable medium of Claim 17, wherein the step of
2 the executable element returning name-value pairs includes the executable
3 element returning a hash table that contains the name-value pairs.

1 19. (original) The computer-readable medium of Claim 13, wherein the
2 computer-readable medium further includes sequences of instructions for
3 performing the steps of:
4 generating a mapping of a plurality of access control functions to digital
5 signatures, wherein the plurality of access control functions include the
6 access control function, wherein one or more classes define an
7 implementation for each of the plurality of access control functions; and
8 wherein each of the one or more classes belong to a superclass.

1 20. (original) The computer-readable medium of Claim 19, further including
2 sequences of instructions for performing the step of invoking a routine defined by
3 a superclass that collects data to return to a caller of the particular class.

- 1 21. (original) The computer-readable medium of Claim 20, wherein the step of
2 executing the executable element includes invoking a routine defined for the
3 superclass.
- 1 22. (original) The computer-readable medium of Claim 13, wherein the step of
2 retrieving an executable element includes retrieving byte code.
- 1 23. (original) The computer-readable medium of Claim 22, wherein the step of
2 retrieving byte code includes retrieving Java byte code.
- 1 24. (original) The computer-readable medium of Claim 13, wherein the step of
2 retrieving an executable element includes a first computer system retrieving byte
3 code transmitted via a local area network from a second computer system.
- 1 25. (currently amended) An access control system, comprising:
2 a processor;
3 a memory coupled to the processor;
4 a first mapping that maps each of a set of access control functions to a digital
5 signature of that access control function;
6 the processor configured to retrieve an executable element in response to a
7 request to execute a first access control function;
8 the processor configured to determine whether the executable element matches
9 the first access control function based on the digital signature; ~~and~~
10 the processor configured to execute the executable element when the executable
11 element matches the first access control function[[]] ; and
12 wherein the set of access control functions are each implemented in a class.
- 1 26. (currently amended) The access control system of Claim 25,
2 ~~wherein the set of access control functions are each implemented in a class; and~~

3 wherein the first mapping maps a class implementing one of the set of access
4 control functions to a digital signature.

1 27. (original) The access control system of Claim 25, further comprising:
2 the processor configured to detect that an access control event has occurred; and
3 the processor configured to retrieve the executable element in response to
4 detecting that the event has occurred.

1 28. (original) The access control system of Claim 27, further comprising:
2 the processor configured to generate a mapping between the access control event
3 and the access control function;
4 the processor configured to determine that the access control event is mapped to
5 the access control function; and
6 the processor configured to retrieve the executable element in response to
7 determining that the access control event is mapped to the access control
8 function.

1 29. (original) The access control system of Claim 28, wherein the executable
2 element returns name-value pairs.

1 30. (original) The access control system of Claim 29, wherein the executable
2 element returns a hash table that contains the name-value pairs.

1 31. (original) The access control system of Claim 25,
2 wherein the processor is configured to generate a mapping of a plurality of access
3 control functions to digital signatures;
4 wherein the plurality of access control functions include the access control
5 function, wherein one or more classes define an implementation for each
6 of the plurality of access control functions; and

7 wherein each of the one or more classes belong to a superclass.

1 32. (original) The access control system of Claim 31, further comprising said
2 processor configured to invoke a routine defined by a superclass that collects data
3 to return to a caller of the particular class.

1 33. (original) The access control system of Claim 32, wherein said processor is
2 configured to execute the executable element by invoking a routine defined for
3 the superclass.

1 34. (original) The access control system of Claim 33, wherein said executable
2 element is byte code.

1 35. (original) The access control system of Claim 34, wherein said byte code
2 includes Java byte code.

1 36. (original) The access control system of Claim 35, wherein said processor is
2 configure to retrieve an executable element by retrieving byte code transmitted
3 via a local area network.